Multiple Relations Classification Using Imbalanced Predictions Adaptation

Sakher Khalil Alqaaidi, Elika Bozorgi and Krzysztof J. Kochut
School of Computing, University of Georgia, U.S.A.

Keywords: Text Mining, Knowledge Graphs, Natural Language Processing, Relation Classification.

Abstract: The relation classification task assigns the proper semantic relation to a pair of subject and object entities; the task plays a crucial role in various text mining applications, such as knowledge graph construction and entities interaction discovery in biomedical text. Current relation classification models employ additional procedures to identify multiple relations in a single sentence. Furthermore, they overlook the imbalanced predictions pattern. The pattern arises from the presence of a few valid relations that need positive labeling in a relatively large predefined relations set. We propose a multiple relations classification model that tackles these issues through a customized output architecture and by exploiting additional input features. Our findings suggest that handling the imbalanced predictions leads to significant improvements, even on a modest training design. The results demonstrate superiority performance on benchmark datasets commonly used in relation classification. To the best of our knowledge, this work is the first that recognizes the imbalanced predictions within the relation classification task.

1 INTRODUCTION

The relation classification (RC) task aims to identify relations that capture the dependency in every pair of entities within unstructured text. The task is employed in several applications, such as knowledge graph construction and completion (Chen et al., 2020) and entities interaction detection in biomedical text (Bundschus et al., 2008). In knowledge graphs, it is common to employ relational triples as the base structure. A triple consists of a subject entity, an object entity, and a semantic relation connecting them. For instance, Wikipedia articles rely on Wikidata knowledge base to provide its content (Vrandečić and Krötzsch, 2014); users can query Wikidata in a structured format using SPARQL and retrieve the information as RDF triples. In biomedical text, the RC task helps in discovering the interactions between entities such as proteins, drugs, chemicals and diseases in medical corpora.

In the supervised RC task, the objective is to learn a function that takes a sentence and its tagged entities as input, then assigns a binary class to each relation from a predefined set. A positive label indicates that the relation is valid for an entity pair. Thus, the output consists of the positive relations. We use this formal notation for the task:

$$f(W,E,P) = \begin{cases} R, & \text{Multiple relations} \\ r, & \text{Single relation} \\ \emptyset, & \text{otherwise} \end{cases} \quad (1)$$

where W is a sequence of words $[w_1,\ w_2\ ...\ w_n]$, E is the set of one or more entity pairs. Each entity pair consists of a subject entity and an object entity, where an entity is a sub-sequence of W. P is the predefined relations set. R is a set of multiple relations found for E. r is a single relation. $\emptyset$ indicates that no relation exists connecting any of the entities. In an example from the NYT dataset (Riedel et al., 2010) for the sentence *"Johnnie Bryan Hunt was born on Feb. 28 , 1927 , in rural Heber Springs , in north-central Arkansas."*, the valid relations are *"contains"* and *"place_lived"*. These relations connect the entities in the pairs *("Arkansas", "Heber Springs")* and *("Johnnie Bryan Hunt", "Heber Springs")*, respectively.

Table 1 shows the average number of relations in two well known benchmarks, NYT (Riedel et al., 2010) and WEBNLG (Zeng et al., 2018). Commonly, a sentence incorporates multiple relations and a single RC approach is only valid for limited cases. However, majority of the literature work follow the single relation approach. Single RC models require additional

Alqaaidi, S., Bozorgi, E. and Kochut, K.
Multiple Relations Classification Using Imbalanced Predictions Adaptation.
DOI: 10.5220/0012455100003636

In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 2*, pages 511-519
ISBN: 978-989-758-680-4; ISSN: 2184-433X

Table 1: The number of predefined relations in the NYT and WEBNLG datasets, the average number of positive relations in each sentence, the standard deviation, and the percentage of sentences with 3 or more positive relations.

Dataset	Relations	Avg.	Stdev.	3+ Rels.
NYT	24	2.00	2.88	18.48%
WEBNLG	216	2.74	2.23	41.72%

preprocessing procedure to be able to identify multiple relations (Wang et al., 2019), that is by replicating the sentence W in equation 1, then assigning an entity pair and a single relation r to each copy. Such approach does not only incur additional steps but also an added training load. An additional downside is losing the contextual information due to splitting the entities data in the input (Qu et al., 2014; Yin et al., 2006), which would result missed accuracy enhancements. Besides that, several single RC models evaluate their work on highly class-imbalanced benchmarks, such as Tacred (Zhang et al., 2017) or datasets with a few predefined relations. For instance, SemEval (Hendrickx et al., 2010) has only six relations. Such performance measurements make it hard to generalize to real-world scenarios. Additionally, these models employ complicated approaches, such as attention mechanisms, additional training and tuning efforts (Wang et al., 2016; Zhou et al., 2016). Furthermore, most approaches neglect the imbalanced prediction pattern in the predefined relations set, when the objective is to predict only one relation out of many others in the predefined relations set.

The multiple RC approach tackles the previously mentioned problems. However, regular methods still unable to achieve competitive results, mainly affected by the need to adapt to the imbalanced prediction. Despite their ability to predict several relations in one sentence, the predicted ones' size is relatively smaller than the predefined relations set in common. This gap is shown in Table 1 when comparing the average number of relations with the predefined set size, which indicates high imbalanced distribution of positive and negative labels in each sentence. Furthermore, the table shows the percentage of sentences of three or more prediction relations, reflecting the importance of the multiple RC task.

In this paper, we propose a **M**ultiple **R**elations **C**lassification model using Imbalanced Predictions **A**daptation (MRCA). Our approach adapts to the imbalanced predictions issue through adjusting both the output activation function and the loss function. The utilized loss function has proved its efficiency in several imbalanced tasks. However, our customization shows additional enhancements within the RC task. Furthermore, we utilize the entity features through concatenating an additional vector to the word embeddings in the text encoder level.

The evaluation shows that our approach outperforms other models that reported their multiple RC performances in the relation extraction task on two popular benchmarks. To the best of our knowledge, this is the first work that addresses the imbalanced predictions within the RC task. The ablation study demonstrates the efficacy of our approach components in adapting to the imbalanced predictions, and in utilizing the text and the entity features. Furthermore, the architecture of our model has a light design that yields astonishing performance. We make our code available online[1].

2 RELATED WORK

2.1 Single Relation Classification

Generally, RC models pursued the approach of generating efficient text representation to identify relations. Early supervised approaches (Wang, 2008; Fundel et al., 2007) employed natural language processing (NLP) tools to extract text features, such as word lexical features, using dependency tree parsers (Klein and Manning, 2002), part-of-speech (POS) taggers and named entity recognition. Relex (Fundel et al., 2007) generated dependency parse trees and transformed them into features for a rule-based method.

With the achievements of neural network methods, deep learning models utilized a combination of text lexical features and word embeddings for the input (Gormley et al., 2015; Zhang et al., 2018) while other approaches (Zhou et al., 2016; Zeng et al., 2014; Lee et al., 2019; Ding and Xu, 2022) depended on those embeddings solely to avoid NLP tools error propagation to later stages (Zeng et al., 2014). Neural network-based models employed word embeddings in different ways. First, embeddings generated from algorithms such as Word2Vec (Mikolov et al., 2013) using custom training data such as in (Gormley et al., 2015; Zeng et al., 2014). Second, embeddings from pre-trained language models (PLMs), such as Glove (Pennington et al., 2014). These PLMs were utilized in the works including (Zhou et al., 2016; Zhang et al., 2018; Lee et al., 2019; Ding and Xu, 2022). In (Zhou et al., 2016), authors presented a neural attention mechanism with bidirectional LSTM layers without any external NLP tools. In C-GCN (Zhang et al., 2018), the dependency parser features were embedded into a graph convolution neural network for RC.

[1]https://github.com/sa5r/MRCA

TANL (Paolini et al., 2021) is a framework to solve several structure prediction tasks in a unified way, including RC. The authors showed that classifiers cannot benefit from extra latent knowledge in PLMs, and run their experiments on the T5 language model.

Bert (Devlin et al., 2018) is a contextualized PLM that has presented significant results in various NLP tasks and several RC models employed it (Wu and He, 2019; Baldini Soares et al., 2019; Cohen et al., 2020; Karaevli and Güngör, 2022). The earliest was R-Bert (Wu and He, 2019), where authors customized Bert for the RC task by adding special tokens for the entity pairs. Later, Bert's output was used as an input for a multi-layer neural network. In (Cohen et al., 2020), the traditional classification was replaced with a span prediction approach, adopted from the question-answering task. In (Karaevli and Güngör, 2022), the model combined short dependency path representation generated from dependency parsers with R-Bert generated embeddings.

2.2 Multiple Relations Classification

Methods that classify multiple relations in a single input pass vary based on the usage of NLP tools, neural networks and PLM models. Senti-LSSVM (Qu et al., 2014) is an SVM-based model that explained the consequences on the performance when handling multi-relational sentences using a single relation approach.

CopyRE (Zeng et al., 2018) is an end2end entity tagging and RC model that leveraged the copy mechanism (Gu et al., 2016) and did not use a PLM. Instead the model used the training platform's layer to generate word embeddings. In the RC part of the model, the authors used a single layer to make predictions over the softmax function. Inspired by CopyRE, CopyMTL (Zeng et al., 2020) is a joint entity and relation extraction model with a seq2seq architecture. The model followed CopyRE's approach in representing text.

Several models employed Bert in the RC task (Wang et al., 2019; Li and Tian, 2020). The work in (Wang et al., 2019) elaborated on the flaws of the single relation prediction in multi-relational sentences and presented a model that is based on customizing Bert. Specifically, the model employed an additional prediction layer and considered the positions of the entities in the input. In (Li and Tian, 2020), authors showed that RC is not one of the training objectives in the popular PLMs. Therefore, they leveraged Bert and used a product matrix to relate the identified relations to the sentence entities.

GAME model (Cheng et al., 2022) used the NLP tool Spacy (Honnibal and Montani, 2017) to generate word embeddings. The model is based on graph convolution networks for global sentence dependency and entities interaction features. ZSLRC (Gong and Eldardiry, 2021) is a zero-shot learning model that used Glove PLM. We mention this work because it reports the supervised learning performance in RC task.

3 METHODOLOGY

Our model incorporates two main components, an output adaptation module and an input utilization module. Between the input and the output modules, we employ neural network with light design to have low training parameters and better performance. We use an average pooling layer to reduce the dimensionality of the network before the output layer. The dropout layer is used to tackle training overfitting. Finally, in the output layer, each unit represents a relation. Figure 1 shows the main architecture of our model.

3.1 Text Encoder

We utilize Glove (Pennington et al., 2014) pre-computed word embeddings to encode the input sentences. Glove embeddings are retrieved from a key-value store where words in lowercase are the keys for a float vectors matrix $R^{s \times d}$, where s is the vocabulary size and d is the embedding dimensions. We find Glove more convenient for the task to tackle the out-of-vocabulary (OOV) (Woodland et al., 2000) problem. Specifically, Glove's most used variant[2] has relatively large dictionary of 400,000 words. However, the embeddings are context-free and the keys are case insensitive. Other popular PLMs have much smaller vocabularies but support Glove's missed features. For instance, Bert (Devlin et al., 2018) generates contextual embeddings and has character case support. Nevertheless, the commonly used Bert variant[3] has 28,997 vocabulary entries only. Thus, the OOV words will get its representation based on the latent training parameters (Nayak et al., 2020). At the same time, several studies showed that RC is not one of the training objectives in Bert (Li and Tian, 2020; Liu et al., 2019). Thus, we adjust Glove to provide the missed features as the following.

First, having case sensitive embeddings is essential to denote entity words in the sentence. Realizing entities in the RC task is crucial to detect the

[2] https://nlp.stanford.edu/projects/glove/

[3] https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4

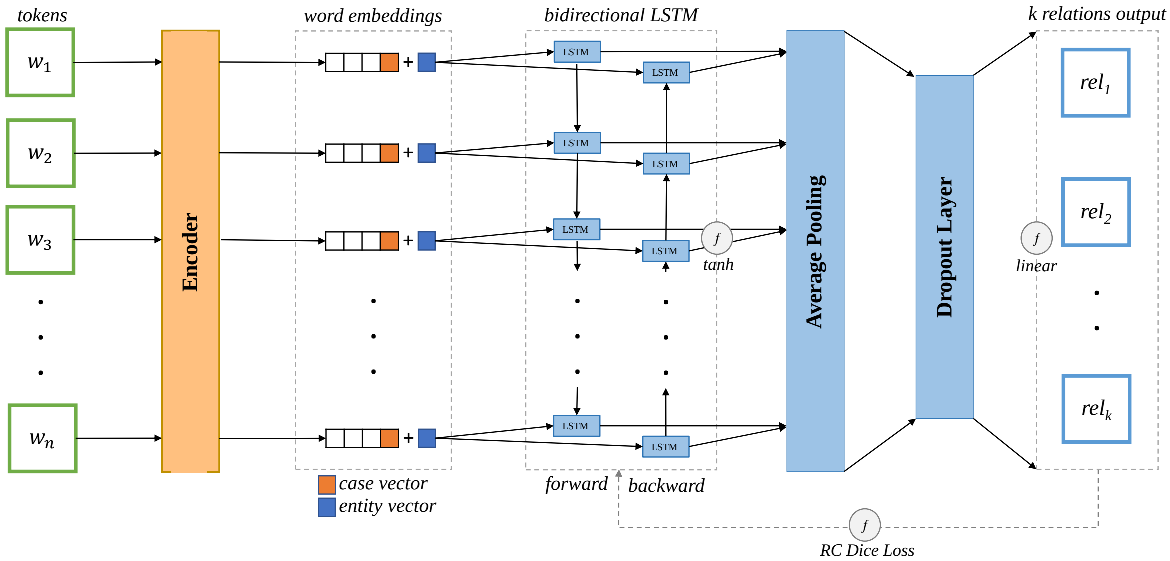


Figure 1: The main architecture of our model. The adaptation approach uses a linear activation function in the output and the Dice loss extension. Furthermore, we enhance the embeddings by adding two vectors, a character case vector and an entity type vector denoted by the orange and blue squares.

proper relation. Generally, a word with an uppercase first character is an entity word. Thus, we add an additional vector to the word embeddings to denote the first character case. For uppercase first character words we use the value of ceiling the largest vector value in Glove. Formally, the vector value is computed as the following:

$$v = \lceil \max_{1 \leq i \leq s} \left(\max_{1 \leq j \leq d} (R[i][j]) \right) \rceil \quad (2)$$

where R is the vectors matrix in Glove, s is the vocabulary size, and d is the embedding dimensions. For lowercase first character words, we use the negative value of v. We employ the maximum and minimum values in the PLM to boost distinguishing entity words from non-entity words. The orange square in Figure 1 denotes the first character case vector.

Second, to provide contextual sentence representation, we make us of a bidirectional long short-term memory (LSTM) as our first layer in the model architecture.

Although we employ a large vocabulary in encoding the sentence, a few words are still not matched. Thus, we generate their embeddings by combining the character level embeddings.

Entity Features. We show in equation 1, that the task input consists of subject and object entities in addition to the sentence. We attempt to enrich the input with these details by following a similar approach of appending an additional vector from the previous step. Specifically, we append a vector of the value v from equation 2 to the word representation when the input indicates that the word is a subject entity or part of it, the negative value of v for object entity words, and 0 for non-entity words. The dense blue square in Figure 1 denotes this additional vector. Formally the vector is given by the function f_{entVec} as the following:

$$f_{entVec}(w) = \begin{cases} v & , \quad w \in E_{sub} \\ -1 \times v & , \quad w \in E_{obj} \\ 0 & , \quad w \notin \{E_{sub} \cup E_{obj}\} \end{cases} \quad (3)$$

where w is a word in the sentence, E_{sub} is the subject entities set and E_{obj} is the object entities set. We use the negative value in the object entity to emphasize the difference between entity types and make the relation direction between entity pairs recognizable while training.

3.2 Imbalanced Predictions Adaptation

In real-world scenarios, the number of relations in a sentence is much smaller than the predefined relations in the RC task. Consider the gap in Table 1 between WEBNLG relations and the average number of valid relations in each sentence. We see that it is impractical to employ traditional probability activation functions in neural networks (NN) for this case. For instance, *sigmoid* and *softmax* are commonly used functions in NNs (Chollet, 2021). Our claim is supported by the fact that these functions treat positive and negative predictions equally. In other words, all probability predictions of 0.5 or greater are considered as positive label predictions in the mentioned functions. Thus, we improve the model's ability to predict negative labels by devoting 75% of the prediction range for the negative labels. We implement this step by restricting the model's layers output to values that vary between -1 and 1. We perform that through applying *tanh* activation function to the first layer, then using a linear activation function in the output layer. As a result, three quarters of the prediction range are allocated for the negative labels, i.e., all predictions between -1 and 0.5 indicate a negative label. Figure 2 compares the prediction ranges in a probability activation function (*sigmoid*) and the output of the *tanh* activation function.

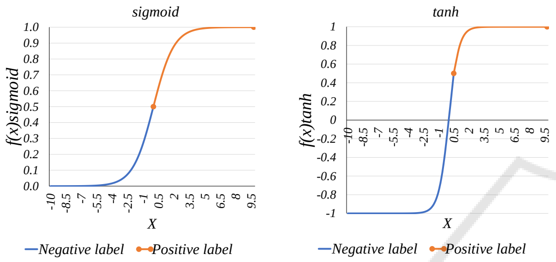


Figure 2: Comparison between prediction ranges in the sigmoid function and our implementation.

Dice Loss Extension. Traditionally, straightforward classification models employ the cross-entropy loss functions (Chollet, 2021), that are used to improve the accuracy, whereas the RC task objective is to reduce the false positive and false negative predictions. Thus, we seek improving the precision and recall performances, i.e., enhancing the model's f1 score. Dice Loss has shown significant results in several domains, such as computer vision (Huang et al., 2018) and other NLP tasks that have imbalanced data (Li et al., 2020). The function was designed with inspiration of the f1 metric as the following:

$$DiceLoss(y_i, p_i) = 1 - \frac{2p_i y_i + \gamma}{p_i^2 + y_i^2 + \gamma} \tag{4}$$

where y_i is the ground-truth label for relation i, p_i is the prediction value, and γ is added to the nominator and the denominator for smoothing, which has a small value of 1e-6 in our implementation.

Utilizing Dice Loss in our adapted predictions may incur unconventional behaviour. Specifically, when having negative ground truth labels and negative value predictions at the same time. Such case would result high loss when using Dice Loss, whereas low loss is the normal result. Our analysis in Table 2 shows the invalid loss values and the expected ones. To address this issue, We expand our adaptation by implementing an extension for Dice Loss. Specifically, we address the negative prediction case by computing the loss from a division operation; the nominator is the squared smoothing value; the denominator is the regular Dice loss denominator. Raising the smoothing value to the second power is necessary to present a small loss value. Our corrected loss value examples can be observed in Table 2. We call this extension, *RC_DiceLoss* and formally defined as the following:

Table 2: Loss calculations for ground truth y and the prediction value p in Dice loss and in our implementation. The underlined numbers are the unconventional values in Dice loss.

y	p	Expected loss	Dice loss	RC Dice loss
0	1	≥ 1	0.9	0.9
0	0.1	≈ 0	$\underline{0.9}$	9e-13
0	-0.1	≈ 0	$\underline{0.9}$	9e-13
0	-1	0	$\underline{0.9}$	9e-13
1	1	0	0	0
1	0	≥ 1	0.9	0.9
1	-1	>1	1.9	1.9

$$RC\_DiceLoss(y_i, p_i) = \begin{cases} \frac{\gamma^2}{p_i^2+y_i^2+\gamma} & , \quad y_i = 0\, and\; p_i < 0.5 \\ 1 - \frac{2p_i y_i+\gamma}{p_i^2+y_i^2+\gamma} & , \quad otherwise \end{cases} \tag{5}$$

4 EXPERIMENTS

4.1 Datasets and Experimental Setup

To demonstrate the generalization and the applicability of our model, we evaluated it on diverse and widely used datasets. The NYT dataset (Riedel et al., 2010) was generated from a large New York Times articles corpus, where each input item consists of a sentence and a set of triples, each triple is composed of subject and object entities, and a relation. WEBNLG dataset was originally generated for the Natural Language Generation (NLG) task, CopyRE (Zeng et al., 2018) customized the dataset for the triples and relations extraction tasks. Table 3 shows the statistics and the splits of the datasets.

Our model achieved the best results using Glove PLM. The language model has been trained on 6 Billion tokens with a 400,000 words vocabulary and 300 dimensional word embeddings. Nevertheless, the experiments demonstrated that our model can adopt other PLMs and still provide competitive results. We performed the experiments using TensorFlow. Our model's hyper-parameters and training settings are

Table 3: Statistics of the evaluation datasets.

Dataset		NYT	WEBNLG
Relations		24	216
Samples	Training	56,196	5,019
	Validation	5,000	500
	Testing	5,000	703
	Total	66,196	6,222

unified for both experimental datasets, which confirms the applicability of our approach to real-world data. Table 4 shows the training settings and the model hyper-parameters. We used Adam optimizer for stochastic gradient descent, and performed the training for five times on every dataset with different random seed and reported the mean performance and the standard deviation. Although we implemented the training for 50 epochs, the mean convergence epoch for the NYT dataset was 21.4. The hyper-parameters were chosen based on tuning the model for best performance. We ran the experiments on a server with NVIDIA A100-SXM-80GB GPU device and AMD EPYC MILAN (3rd gen) processor, but using only 8 cores. We used only 20GB of the available main memory for the WEBNLG dataset experiments and 100GB for the NYT dataset due to its size. We conducted an ablation study to test our model's components using different variants as shown in Section 4.4.

4.2 Comparison Baselines

We compare our results with the following supervised models. We refer to the main characteristics of each one in section 2. CopyRE (Zeng et al., 2018) and CopyMTL (Zeng et al., 2020) are based on the copy mechanism and used the same approach to generate word embeddings. Both evaluated their work on the NYT and WEBNLG datasets. GAME model (Cheng et al., 2022) used Spacy to generated word embeddings and reported their results on the NYT dataset.

Table 4: Model hyperparameters and training settings.

Parameter		Value
Average Pooling	Pool Size	80
	Strides	2
Learning	Rate	0.0015
	Decay	3e-5
Bi-LSTM units		2×500
Dropout rate		0.15
Sequence padding		100
Epochs		50
Early stopping patience		5
Batch size		32
Generated parameters		13M
Average epoch time		2355ms

Table 5: Our model's F1 scores and the standard deviations (subscripts) on the NYT and WEBNLG datasets compared with the baseline models.

Model	GAME	CopyRE	CopyMTL	MRCA
NYT	77.1	87.0	86.9	$\mathbf{96.65}_{0.17}$
WEBNLG	-	75.1	79.7	$\mathbf{93.35}_{0.29}$

Other multiple relations classification models were not considered in the comparison due to their utilization of a different release of the NYT dataset, such as (Li and Tian, 2020) and ZSLRC (Gong and Eldardiry, 2021). We found that the used release is not commonly used in the literature.

4.3 Main Results and Analysis

We report our average F1 scores in Table 5 for the NYT and WEBNLG datasets, respectively. Additionally, we visualize the training performance in Figure 3. The results show superiority among the baseline models. We report the precision and recall scores in Table 6. We highlight our results in the WEBNLG dataset, as we find that relation predictions in that dataset is highly imbalanced due to the large number of predefined relations. Furthermore, the dataset has smaller training data. Nevertheless, the WEBNLG's F1 score is close to the NYT's score. Knowing that, the NYT dataset has much smaller predefined relations and more training data, which indicates that our adaptation method supported achieving better predictions despite the imbalanced distribution of the binary labels.

4.4 Ablation Study

To examine the effectiveness of our model's components, we evaluate the imbalanced predictions adaptation approach, and the text encoder adjustments. We design different variants of our model and perform training using the same main evaluation settings in Table 4. Moreover, We report the average score of five runs and the standard deviation. We use the WEBNLG dataset for the ablation study experiments. We report the performances in Table 6, then we present the following analysis.

Imbalanced Predictions Adaptation Effectiveness. To evaluate the contribution of our imbalanced predictions adaptation approach, we assess our model using different activation and loss functions. Specifically, we use the traditional *sigmoid* activation function and the binary cross entropy loss function. We report this variant's performance in Table 6 with the

Table 6: The performance of our model's variants on the WEBNLG dataset. The subscripts show the standard deviation.

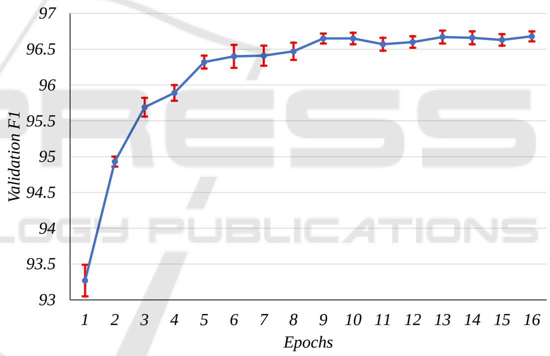
Model	Precision	Recall	F1
MRCA	$95.4_{0.25}$	$91.3_{0.48}$	$93.35_{0.29}$
MRCA-Sigmoid-BCE	$93.35_{0.31}$	$88.73_{0.55}$	$90.88_{0.3}$
MRCA-Bert	$94.5_{0.2}$	$89.9_{0.49}$	$92.15_{0.26}$
MRCA-Bert-noLSTM	$55.18_{2.21}$	$53.7_{1.1}$	$54.4_{1.16}$

name *MRCA-Sigmoid-BCE*. The variant's F1 score is approximately 3% less than our model's score, which is an average value between the precision scores difference and the recall scores difference. Noting that the recall gap is larger, which presents the first indication that the adaptation approach improved predicting negative labels.

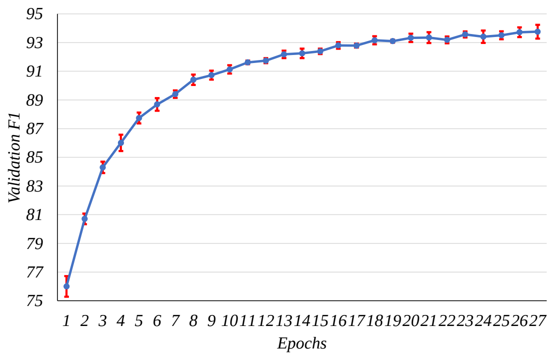
Encoder Effectiveness. To evaluate our text encoder adjustments, we need to consider two subcomponents in the assessments, that are the usage of Glove language model and the addition of the entity type vector to the embeddings. Thus, we test the following variants of our model. *MRCA-Bert* is a variant that uses Bert PLM instead of Glove and *MRCA-Bert-noLSTM* is a variant that uses Bert but with no LSTM layers. We use Bert's release[4] with character case support since we added the same case feature in our implementation. In the former variant, there is a slight difference between the reported F1 score and our model' score, which demonstrates less contribution of the Glove employment in our overall performance. However, using Glove, our model still outperforms the Bert variant due to the better OOV terms support. Noting that Bert is known as a language model with contextual text representation support. Thus, the assumption is that, the LSTM layers would not affect Bert's performance. Nonetheless, in the second variant *MRCA-Bert-noLSTM*, the performance is way worst. This result supports our claim that RC is not one of Bert's training objectives in section 3.1 because of the abstract usage of Bert. Furthermore, with a weak contextual representation in Bert, OOV words will split into non-meaningful tokens as described in the tokenization algorithm that is used in Bert (Song et al., 2021). This concludes the importance of using a language model with larger vocabulary.

5 CONCLUSION

We propose MRCA, a multiple relations classification model that aims at improving the imbalanced predictions. Our light-design implementation leverages wider prediction range for negative labels and customize a remarkable loss function for the same purpose. Furthermore, text and entity features are utilized efficiently to improve the relations prediction. The experiments presented superiority among state-of-the-art models that reported the relation classification performance. Assessing our model's components showed that addressing the imbalanced predictions yields significant improvement in the relation classification task. Furthermore, representing sentences using language models with rich vocabularies provides performance enhancements in the relation classification task.



(a)



(b)

Figure 3: The validation F1 score during training for the evaluation datasets. (a) indicates the NYT training performance. (b) indicates the WEBNLG training performance.

[4]https://tfhub.dev/tensorflow/bert_en_cased_L-12_H-768_A-12/4

6 FUTURE WORK AND LIMITATIONS

Although the relation classification task has limited applications as a single module, it has wider usages in the relation extraction task. Therefore, we see that our approach can be adopted to achieve new scores in several applications that utilize the relation classification task. Further improvements can be achieved when using NLP tools for lexical and syntactic text features. Additionally, it would be typical to advance our model to assign the predicted relation to the corresponding entities pair in the input. However, this approach cannot be considered as an ideal way for the relation or triple extraction task because errors in the entities tagging step would propagate to the relation classification task. Finally, our imbalanced predictions adaptation promises enhancements if used by similar tasks of imbalanced classes.

Our evaluation was limited by the small number of models that reported the relation classification performance. However, the results proved our model's superiority, denoted by the gap between our F1 score and the closest model.

REFERENCES

Baldini Soares, L., FitzGerald, N., Ling, J., and Kwiatkowski, T. (2019). Matching the blanks: Distributional similarity for relation learning. In *ACL*, pages 2895–2905. Association for Computational Linguistics.

Bundschus, M., Dejori, M., Stetter, M., Tresp, V., and Kriegel, H.-P. (2008). Extraction of semantic biomedical relations from text using conditional random fields. *BMC bioinformatics*, 9(1):1–14.

Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., and Duan, Z. (2020). Knowledge graph completion: A review. *Ieee Access*, 8:192435–192456.

Cheng, H., Liao, L., Hu, L., and Nie, L. (2022). Multi-relation extraction via a global-local graph convolutional network. *IEEE Transactions on Big Data*, 8(6):1716–1728.

Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster.

Cohen, A. D., Rosenman, S., and Goldberg, Y. (2020). Relation classification as two-way span-prediction. *arXiv preprint arXiv:2010.04829*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, H. and Xu, X. (2022). Relation classification based on selective entity-aware attention. In *CSCWD*, pages 177–182. IEEE.

Fundel, K., Küffner, R., and Zimmer, R. (2007). Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Gong, J. and Eldardiry, H. (2021). Zero-shot relation classification from side information. In *CIKM*, pages 576–585.

Gormley, M. R., Yu, M., and Dredze, M. (2015). Improved relation extraction with feature-rich compositional embedding models. *arXiv preprint arXiv:1505.02419*.

Gu, J., Lu, Z., Li, H., and Li, V. O. (2016). Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., and Szpakowicz, S. (2010). SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38. Association for Computational Linguistics.

Honnibal, M. and Montani, I. (2017). spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420.

Huang, Q., Sun, J., Ding, H., Wang, X., and Wang, G. (2018). Robust liver vessel extraction using 3d u-net with variant dice loss function. *Computers in biology and medicine*, 101:153–162.

Karaevli, H. A. and Güngör, T. (2022). Enhancing relation extraction by using shortest dependency paths between entities with pre-trained language models. In *INISTA*, pages 1–7.

Klein, D. and Manning, C. D. (2002). Fast exact inference with a factored model for natural language parsing. *Advances in neural information processing systems*, 15.

Lee, J., Seo, S., and Choi, Y. S. (2019). Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing. *Symmetry*, 11(6):785.

Li, C. and Tian, Y. (2020). Downstream model design of pre-trained language model for relation extraction task. *arXiv preprint arXiv:2004.03786*.

Li, X., Sun, X., Meng, Y., Liang, J., Wu, F., and Li, J. (2020). Dice loss for data-imbalanced NLP tasks. In *ACL*, pages 465–476. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nayak, A., Timmapathini, H., Ponnalagu, K., and Venkoparao, V. G. (2020). Domain adaptation challenges of bert in tokenization and sub-word representations of out-of-vocabulary words. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 1–5.

Paolini, G., Athiwaratkun, B., Krone, J., Ma, J., Achille, A., Anubhai, R., Santos, C. N. d., Xiang, B., and Soatto, S. (2021). Structured prediction as translation between augmented natural languages. *arXiv preprint arXiv:2101.05779*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Qu, L., Zhang, Y., Wang, R., Jiang, L., Gemulla, R., and Weikum, G. (2014). Senti-lssvm: Sentiment-oriented multi-relation extraction with latent structural svm. *Transactions of the Association for Computational Linguistics*, 2:155–168.

Riedel, S., Yao, L., and McCallum, A. (2010). Modeling relations and their mentions without labeled text. In *ECML PKDD*, pages 148–163. Springer.

Song, X., Salcianu, A., Song, Y., Dopson, D., and Zhou, D. (2021). Fast WordPiece tokenization. In *EMNLP*, pages 2089–2103.

Vrandečić, D. and Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Wang, H., Tan, M., Yu, M., Chang, S., Wang, D., Xu, K., Guo, X., and Potdar, S. (2019). Extracting multiple-relations in one-pass with pre-trained transformers. In *ACL*, pages 1371–1377. Association for Computational Linguistics.

Wang, L., Cao, Z., De Melo, G., and Liu, Z. (2016). Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307.

Wang, M. (2008). A re-examination of dependency path kernels for relation extraction. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Woodland, P. C., Johnson, S. E., Jourlin, P., and Jones, K. S. (2000). Effects of out of vocabulary words in spoken document retrieval. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 372–374.

Wu, S. and He, Y. (2019). Enriching pre-trained language model with entity information for relation classification. In *CIKM*, pages 2361–2364.

Yin, X., Han, J., Yang, J., and Yu, P. S. (2006). Efficient classification across multiple database relations: A crossmine approach. *IEEE Transactions on Knowledge and Data Engineering*, 18(6):770–783.

Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. (2014). Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344.

Zeng, D., Zhang, H., and Liu, Q. (2020). Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. In *AAAI*, volume 34, pages 9507–9514.

Zeng, X., Zeng, D., He, S., Liu, K., and Zhao, J. (2018). Extracting relational facts by an end-to-end neural model with copy mechanism. In *ACL*, pages 506–514.

Zhang, Y., Qi, P., and Manning, C. D. (2018). Graph convolution over pruned dependency trees improves relation extraction. *arXiv preprint arXiv:1809.10185*.

Zhang, Y., Zhong, V., Chen, D., Angeli, G., and Manning, C. D. (2017). Position-aware attention and supervised data improve slot filling. In *EMNLP*, pages 35–45.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., and Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 207–212.